DOCUMENT RESUME

ED 134 159 IR 004 318

AUTHOR Sorensen, Bettye D.
TITLE A Total Concept of Computer-Based Instruction.
INSTITUTION Boeing Computer Services, Inc., Seattle, Wash.
PUB DATE Aug 76
NOTE 24p.; Paper presented at Association for Development of Computer-Based Instructional Systems (Minneapolis, Minnesota, August 10-12, 1976)

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS Audiovisual Aids; Autoinstructional Aids; *Autoinstructional Programs; Branching; *Computer Assisted Instruction; Computer Graphics; *Computer Oriented Programs; Educational Technology; Input Output; *Instructional Technology; On Line Systems; Program Costs; Programed Instruction; Systems Development
IDENTIFIERS Boeing Computer Services

ABSTRACT

Computer-based instruction (CBI) encompasses the functions of computer administered and managed instruction supplemented by utilities and interfaces. The course author's function in computer-administered instruction (CAI) is to construct the lesson for input in one of three formats: (1) batch entry used by those with knowledge of programing language, (2) user- initiate mode directly typed into the file, or (3) interactive coaching/prompting, a dialogue format for those without programing knowledge. To aid the author, answer analysis techniques, branching, student aid and task interruption controls, editing, and lesson formats are available. Students are preassigned or given free access to lessons. Computer-managed instruction (CMI) involves human management using tools provided by CBI--i.e., required data, assignment of system users, catalog construction, message capabilities, report control and lesson verification. The utilities of flowcharting, the calculator, and automated report generation language (ARGL), internal and external interfaces of hardware, and graphic and audiovisual aids enhance the capabilities of the system. Factors to consider in determining the impact of this CBI system are cost, installation core requirements, and acceptability by the user. Descriptions of answer analyses and standard system reports are appended. (KP)

A TOTAL CONCEPT OF COMPUTER-BASED INSTRUCTION

Bettye D. Sorensen

August 1976

Education and Training Division
Boeing Computer Services, Inc.
Seattle, Washington

A TOTAL CONCEPT OF COMPUTER-BASED INSTRUCTION

INTRODUCTION

Several years ago, when those of us in education and training were first considering the computer as an instructional tool, we were exposed to terms such as computer-assisted instruction, computer-aided instruction, and computer-administered instruction. Then a few years later came computer-managed instruction. And even terminal-managed instruction. Why so many terms, as well as ill-defined terms? Well, we were searching -- searching for the functional niche into which the computer should fit as a part of the instructional process.

We are now beginning to identify that niche. Even to the point that the original "fears" instructors and teachers had about the big machine "replacing" them is beginning -- and notice I'm saying "beginning" -- to dissipate.

So now we have come up with a new term, computer-based instruction. This term, frequently shortened to CBI, is handy. It allows us to define many types of instructional systems which use the computer in <u>any</u> way as computer-based instructional systems.

My purpose in this presentation is to define for you what we at Boeing Computer Services believe should be the <u>total</u> concept of computer-based instruction. Without this total concept -- that is, total utilization of the computer tool -- we will fail to use this instructional asset to its fullest capability.

At this point, I will brief you on the background of the CBI system development on which this presentation is based. For several years, Boeing Computer Services has had a prototype CAI system in operation internally. This prototype system consists of the capability for a student to interact with pre-stored instructional material, reading the material and responding to prompts and/or questions presented via the computer terminal, and then branching according to the response made by the student. This feature is not much different than most interactive computerized training systems. However, the unique aspect of this prototype system is the capability for an instructor to enter his material into the system <u>without</u> -- and I emphasize the word without -- having to know code words, a programming-type language, or even a structured format for entry of the material. The instructor (that is, the course author) responds to computer prompts and cues just as the student does in the training mode.

In 1974, we began to look at this prototype system to determine its potential as a total CBI system. To put a lot of effort into a few words, we researched the capabilities and techniques offered by other computerized instructional systems. We studied the past experience of our own education and training personnel who had been associated with computerized training systems and their applications. And, we analyzed the current and projected needs in education and training methodologies. As a result, we have designed and are in the construction stage of a true computer-based instruction system, one that comprises the total concept of CBI.

OVERVIEW OF THE FUNCTIONAL SYSTEM DESIGN

The graphic representation in Figure 1 gives an overview of the functional design of our new system. To explain to you our total concept of CBI, I will use this functional design of a system -- a system that I can use as I feel necessary in my role as the instructor, or which I can use as I feel necessary to manage the instructional process, or which I can use as a student to obtain individualized instruction.

You will note that I address this CBI system from a functional standpoint. Why? Because only through the completeness of the functional capability of the teaching and the instructional management process can the CBI system meet the needs of this type of approach to education and training.

Functionally speaking, the total CBI system needs

- The capability to administer instruction, interactively, to a student.
- The capability to totally manage all instruction, including computer-administered and traditionally-administered instruction, as well as operations directly or indirectly affecting the instructional process.
- Utilities which may be used selectively to enhance administration or management of instruction, and
- Flexibility to adapt to the individual user's needs.

Let's briefly examine each of these functions.

COMPUTER-ADMINISTERED INSTRUCTION (CAI)

The capability to administer instruction, interactively, to a student is referred to in a total CBI system as the CAI function, that is, the Computer-Administered Instruction function. This element of the CBI system addresses two type of system users: the student and the instructor (or as we shall refer to the latter, the course author).

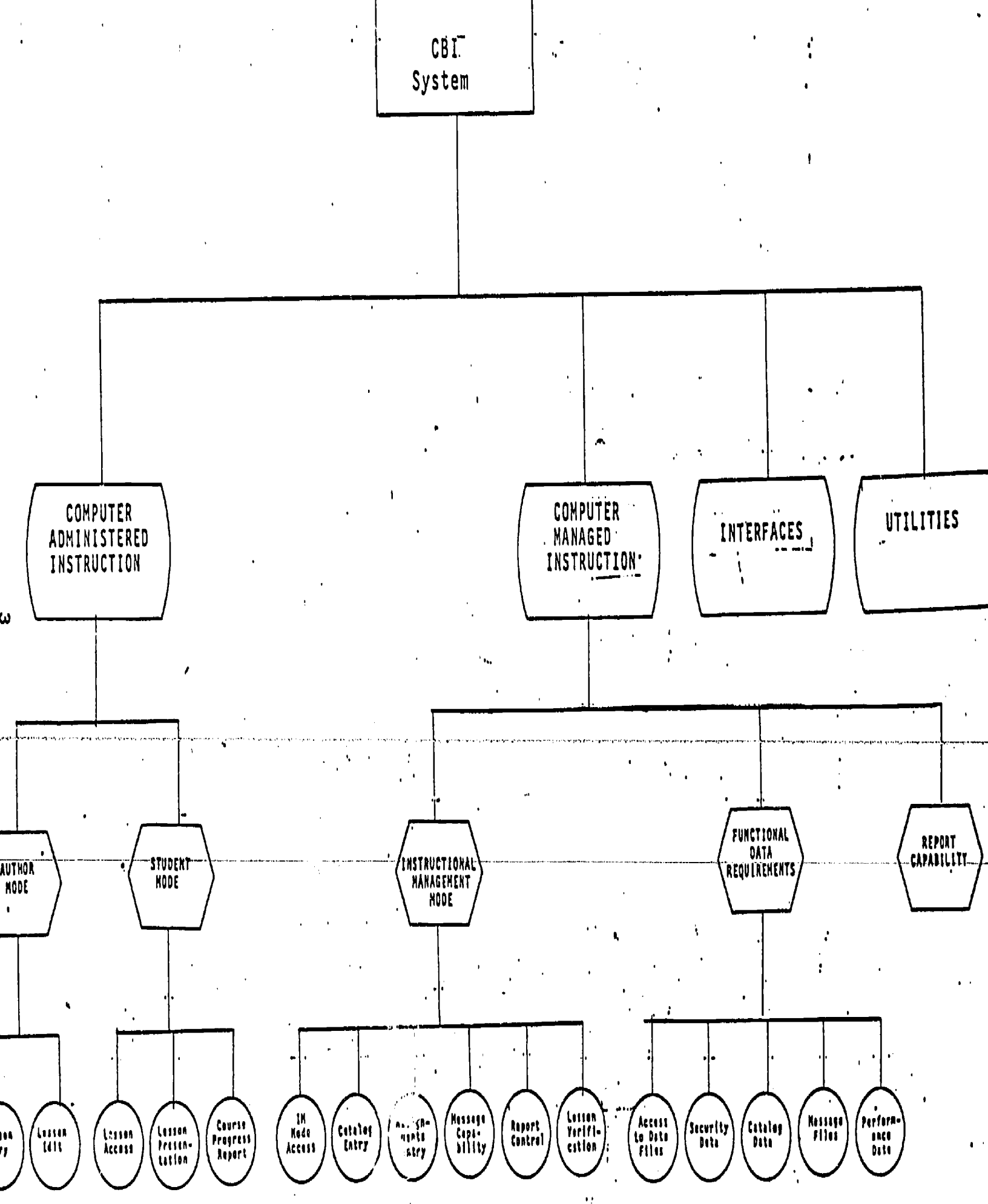

Figure 1. CBI Functional System Design

Author Mode

The author's main function when using the CBI system is to construct lesson material in such a way that the transfer of learning to the student is accomplished. In the CBI system, before the author can do this task, he/she is assigned by the management function to perform the function of lesson construction. Notice the overlap in the two functions of instructing and managing. Later in this presentation I will address the assignment function as actually a part of the management of the total CBI system. Because of the diversity of applications of instruction and/or diversity or organization of the system user which may or may not exist, the assigned author will be allowed to construct and/or alter only the designated courses as determined by the management function.

This concept of the CBI system will allow entry of lesson material to be adaptable to the instructor/author and the particular environment in which he prefers to accomplish his function. Consequently, the CBI system will allow lesson entry in a variety of ways:

- In batch format, either from an off-line input device or from the terminal.
- In user-initiated format from the on-line terminal, with the author entering the material using a simple set of format parameters and with little guidance from the system.
- In interactive coaching/prompting format, with the author responding to computer-initiated dialogue and entering the material in totally free format.

The batch entry format allows course authors with knowledge of programming language techniques to build their material accordingly. The author, using a simple set of commands, builds a lesson on a sheet of paper or a coding sheet or some similar form. Then the lesson is keypunched and read into the computer, using the standard JCL required by the user's computer system. The author may generate and verify the lesson material totally through the offline batch facilities. If the terminal is used for batch entry, the statements are treated the same as if entry were made from an off-line device.

When the material is generated, the author will receive a printout of his lesson with appropriate diagnostic messages for any incorrect material type or unsatisfied branch points. The author then resubmits the corrected material until the lesson is completely verified.

For the more experienced author, that is, one who has used a particular system for some length of time and is very familiar with the entry process, we have designed the user-initiated online entry format. This technique also allows a more sophisticated clerical type to enter the lesson material for the course author, thus freeing the instructor's time from this function.

The user-initiate entry mode is similar to the batch entry mode, except that the author has more freedom in formatting material and the material is typed directly into the computer files from the hardcopy or CRT terminal. In this mode, any error messages are directed back to the author via his terminal and he then has an opportunity to correct his errors and to immediately reverify his lesson without signing off his terminal.

The third method of entry, interactive coaching/prompting mode, allows the author to enter the lesson material without having any knowledge of computer-type codes, format parameters, or other such restrictions. The author responds to the computer-initiated dialogue and enters the material accordingly. Two degrees of cueing are incorporated into this method, "loud" -- in which detailed cueing is presented, and "whisper" -- in which abbreviated cueing is presented.

As the author develops and enters the instructional material using this entry mode, it is automatically verified for incomplete branches or unanswered questions. The system will not allow the author to complete the lesson until all branch points within the lesson structure have been satisfied.

Answer Analysis Techniques

In addition to the versatility in entry techniques for instructional material, the CBI system must also be versatile in answer analysis techniques. The instructor must be able to analyze students' responses in much the same way as is done in the classroom setting. In defining the structure for these techniques, we found that they fall into two basic categories: character analysis or word analysis. To these basic categories may be added one or more qualifiers. Therefore, in the design of the CBI system, the combining of multiple qualifiers to either the character or word answer type analysis makes more than fifty (50) different answer techniques possible. A brief description of the basic answer techniques and the qualifiers is presented in Appendix 1.

Branching Control

Under author control, several types of branching may be used in the CBI system: conditional, unconditional, subroutine calls, and implicit branching. In unconditional branching, the author directs the system to branch the program directly to some other point when the end of the frame is reached. In addition to expressing an unconditional branching point, the author may select to repeat a block of text. Or, he may choose to fill in the branch point at a later time (but prior to verification of lesson material).

Conditional branching is set up by the answer analysis technique selected for use by the author. Depending on the technique selected and the response a student gives to a cue, the author may direct the program to a new block of text or to a previous block, or to the same block.

The CBI system also incroporates a type of subroutine call. Rather than set up a branch point to and a return point from a subroutine in some other lesson material within the system files, this subroutine call causes the selected material to be copied into the lesson being created or modified.

The author may also designate a possible set of implicit branches. This is accomplished by providing a list of anticipated correct and/or incorrect answers with accompanying branch points. The author also designates branch points to be taken in case of unanticipated answers. The author has absolute control over the number of times the student may try to respond to the cue.

Lesson Formatting

The author has a choice of four functional types of frames (or blocks) to use within the lesson: tutorial, performance, test, and control. Material constructed in the tutorial block format imparts information to the student and does not require a response from the student. At the end of the frame, the author must designate an unconditional branch point.

The performance block may contain text and/or cueing, but it requires some type of response from the student. The response analysis and associated branching is designated by the author.

The third functional block type is the test block. The author, in the total concept of a CBI system, must be allowed versatility in placement of tests within the instructional material. Therefore, this CBI system provides the capability for the author to build a pretest, a posttest, or to place the testing activity within the lesson itself. Another feature available for building tests is random test item selection. That is, the author is allowed to store a number of questions within the test file and then designate random selection of a certain number of those items for presentation to the student. Within a test item itself, as well as within a performance item, the course author is allowed to build multiple variables within the problem statement and to designate random selection of those variables for presentation to the student.

Control of Student Aids

Within the parameters of lesson construction, the author has absolute control over aids which the student may use to maximize the transfer of learning. For example, the author has the capability to build

definitions, additional explanations, or resource references within the lesson framework to which the student may refer when and if he needs assitance.

In addition, the author may provide messages to an individual student or to a group of students for presentation when the student signs on to take the lesson. And the author may control various audio/visual equipment in support of the lesson material, depending upon the equipment available to a particular user's system. I will address the types of equipment that may be interfaced to the CBI system later in this presentation.

Another specific type of training aid which the author may control is the calculator. The author designates availability of the calculator to the student in building the lesson. And the author may choose to use the calculator to determine answers to problems given to the student or in controlling the branching parameter as designated by the author.

Editing Capability

The interactive editing feature within the CBI system is used by the author, as assigned by the management function, to modify, add, or delete lesson material or to change instructional strategy. Three types of edit commands are used in the system: those which locate specific material within the system files, those which modify material within the system files, and those which extract material from the system files. For example, a locate type of function would be a command such as "FIND". A modify function would be one such as "CHANGE", or "DELETE". And an extraction type of function would be one such as "PRINT".

The system has been designed to prevent careless modification to system files that would result in excessive time and effort in maintaining system materials.

Task Interruptions

Another item to be addressed in the development of the total concept of a CBI system is control of activity when an unforeseen circumstance occurs, such as a system breakdown. One of the problems that has been faced by many computerized instructional systems is the duplication of effort required on the part of the instructor when something unforeseen occurs. The total CBI system must account for these possibilities and provide a means for eliminating the duplication of effort on the part of either the instructor or anyone else who happens to be using the system when an unforeseen circumstance occurs. The system I am describing provides the capability to save all material at frame level, so that when the author is entering lesson material and an interruption occurs, either system generated or author generated, the previously stored material is saved to the point of the previous frame.

Summary of Author Role

To summarize the author function within the CBI system, we must address the versatility which the instructor/author needs to construct the material according to disciplines required by the environment in which he/she works. The total concept of the CBI system provides this versatility and emphasizes the role the author plays in the successful use of a CBI system.

Student Mode

The second function of the CAI portion of the CBI system is that of the student. The student user takes the courses which have been generated by the course authors and stored on the sytem files.

Controlled student access to lesson material is a necessity in the CBI system. Cost-effectiveness of training requires control over all facets of system activity, including student activity. However, the total concept of the system provides adaptability to individual user's situations. Therefore, this system provides student access to course material in one of two ways: free access and preassignment. The access method is designated by the user installation, not by the constraints of the system itself.

If the user installation chooses to allow free access to course material, anyone may sign onto the system and take courses. In this event, a list of free courses must be provided for the user to choose from.

For a student to be preassigned to course material, an instructional manager must have previously entered the student's name and identification into the system files. Otherwise, the student cannot gain access to the system. An assignment to specific courses is also made for each student. The courses may be assigned in one of two ways, structured, or unstructured. If a structured course assignment is made, the student is required to take the course in the order designated by the assignment. If an unstructured course assignment is made, the student may take the courses in any order he/she chooses.

Student Performance

As the student progresses through a lesson, his performance is recorded. The student's path through the lesson material is determined by the responses he/she makes in relation to the branching patterns built into the material by the author. In this manner, the lesson can be individualized to each student. The student can also request computational, dictionary, and reference assistance.

When a student requests help and there is no information on file for the item requested, that item will be stored in the system files as an unresolved request. The lesson author can then review these files later and put the information into the system if deemed necessary.

Personal Assistance

To provide the student with a direct communication link to the instructor, the CBI system provides the student with the capability to ask direct questions of the instructor/author and to receive the instructor's answers. I have previously addressed the capability of the author to enter direct messages to a single student or to an entire group of students within the lesson framework.

COMPUTER-MANAGED INSTRUCTION (CMI)

The second major element in the total concept of a computer-based instructional system is that of management. All instructional processes require management, whether it be the scheduling of traditional classroom classes, the administration of PI text books, or the use of the computerized instructional system. In the past, much of the control of the instructional process has been provided by manual methods, even though the administration of the instruction might be via CAI. The total concept of the CBI system requires that management of the instructional process be accomplished via the computer.

The management function involves three aspects: (1) actual control over the instructional process; (2) the collection and storage of data related to the instructional process; and (3) the extraction of that data.

Instructional Management

For a comprehensive management approach, each CBI installation should have an instructional manager. I refer to the function of instructional manager within the total CBI system. This function may be performed by one or more persons, depending on the size and the applications of the installation. If more than one person performs the management function, they will be organized in the system files in a hierarchical manner.

The instructional manager has the responsibility for the following functions:

1. The assignment of students and the courses which they are to take.

2. The assignment of courses to authors for purposes of generating and/or maintaining instructional material.

3. The assignment of other instructional managers who will report to the assigning manager.

4. Maintenance of an inventory of equipment and the scheduling of that equipment.

5. Maintenance of system security.

6. Structuring and maintaining the catalog of courses.

7. Maintenance of system files.

8. Control of system reports concerning all aspects of system activity (student progress, system utilization, cost data, etc.)

The CBI system must provide the instructional manager with the tools needed to perform all of these functions within the confines of the automated system.

In our design of the CBI system, we have classified the subfunctions of the management process basically as: (1) access to the management functions; (2) assignment of system users, including course assignments and functional assignments; (3) catalog construction; (4) message capability; (5) report control; and (6) lesson verification. You will note that some of these functions have already been addressed in the other elements of the CBI system. This illustrates the interaction of the components of this total approach to a computerized training environment.

Controlled Management Access

The functions listed above may indicate a lot of activity and responsibility to place on the shoulders of the instructional manager. However, the adaptable CBI system provides for sophisticated management organization only if the user application requires that sophistication. The management function within the CBI system can be structured in a very simple organization if required. Again, adaptability to the user's individual situation is the keynote.

But whether the instructional management function is represented by one person or by many persons within the user installation, a logical hierarchical structure exists. And even within the management function, control must be exercised on the system user. This control permeates the hierarchical organization of the tasks to be accomplished.

To accommodate an adaptable management hierarchy within the system, functionally it is necessary for one management role to assume authority. We call this role the instruction manager. This role can then provide the breakdown into the required sub-roles. Access to any of these subfunctions is controlled by previous assignment to that role within the system.

Assignment of System Users

Control of access to certain elements within the system is one of the required management features. Why? Cost effectiveness has been one of the prohibitive factors in CAI systems in the past. We must address this factor if we are to produce a successful CBI system. In analyzing the causes of "less-than-cost effectiveness", we have determined that lack of control over system access has been a main inhibitor to the use of computerized instructional systems.

In this CBI system, the user in any role must be assigned to that role. The method of assignment is defined, however, by the user installation. For example, in a large and/or highly controlled environment, the assignment function may be highly structured, with each user having a unique assignment code. On the other hand, if the user installation prefers, blanket assignments can be made at any level of control necessary within the organization.

The assignment function involves both assignment of system users and scheduling of students to courses within the system. Again, the sophistication of course assignments may be at any level required by the user.

Catalog Construction

The organization of lessons into courses, and courses into other courses, is another function of management. The catalog file structure within the system is designed so that this structure can be easily established and/or modified. The catalog control can be established to relate also to the instructional management structure within the user organization.

In this system, the catalog files can also be used as an inventory control for equipment and other facilities. Hierarchically organized control can be exercised, if required.

Message Capability

The management function has the capability to send messages within the system. These messages may be addressed to all users signing onto the system, users signing into specific functions, or to individual users.

Report Control

Report control is another function of the management process. Either standard reports, constructed automatically by the system, or user-defined reports may be requested. These types of reports will be discussed later.

Lesson Verification

Though lesson verification is performed in relation to lesson construction, it is a management function. Verification is performed either automatically, in interactive coach/prompt entry mode, or as a result of command by the author mode or management mode.

Data Requirements

In designing the total concept of the CBI system, the functional data requirements involve that data which is necessary for total management of the instructional process. Data types such as security data, catalog file data, message files, and performance data must be considered. The performance data may involve information that is collected and/or computed automatically by the CAI process, or it may be performance data which is entered by the instructional manager and related to traditionally-administered instruction (that is, instruction administered by some media other than the CAI function). Design of the CBI system makes available to the user any and all information needed to successfully perform and control the instructional process within a specific instructional environment.

Report Capability

Reporting is certainly an important function in the management process. Two methods are available in the system for obtaining reports: standard reports, automatically produced upon call by system-contained controls, and user-defined reports, produced by the user with a simple-to-use report generation language. Appendix 2 lists the types of standard reports available from the system.

In defining specialized report content and format, the manager uses a simple set of English-like statements. The system calculator may also be used in developing the report contents. A user-defined report may then be stored as a standard report format within the user's system, if desired.

UTILITIES

Certain features of the CBI system have been categorized as utilities. These features are used to enhance the capabilities and operations of elements within the system.

For example, the flowcharting feature may be used by the author or manager to draw a verbal map of the instructional paths available within a lesson. This feature shows which blocks branch to other blocks. It also indicates whether the lesson is able to terminate, and it produces a formatted frame-by-frame printout of the lesson. Finally, the flowcharter can produce, optionally, a complete concordance for the lesson or a cross-reference for a particular word within the lesson.

The calculator is a system utility that is widely used throughout the other system elements. The computational ability of this part of the CBI system is about the same as most high-level computer language calculators. In addition to ADD, SUBTRACT, MULTIPLY, and DIVIDE, a number of functions are available, such as absolute value, square root, sine, cosine, tangent, and exponentiation. Relational comparison is also available (e.g., equal to, less than, greater than, and combinations thereof). When integer arithmetic is performed, the results may be truncated or rounded, as specified by the user.

If allowed by the lesson author, the calculator may be used by the student within the lesson activity. It may also be used by the author to calculate answers to problems given to the student, or to control branching in course paths. Or, as previously stated, it can be used in report definition.

A third utility with the system is an easy-to-use report generation language. We refer to this language as ARGL, an automated report generation language. It is a report generation language designed to allow authors and managers to dynamically control student paths through lessons and to generate reports from the statistics kept by the CBI system. By using ARGL in the lesson, an author may generate random numbers, calculate values, print variable data, and check variables for branching conditions. With this tool the author can construct frames that evaluate the student's progress and cause the student to branch ahead in a lesson or branch to remedial frames, depending on the student's past performance. The ARGL language can be used by the instructional manager to generate reports using the statistics provided by the system. ARGL has been kept as simple as possible and requires little training to master. Yet, it has also been designed to accommodate sophisticated programming techniques, such as the new technique of structured programming. It, therefore, is a very versatile report generation language.

One may question that, by providing interface for such a variety of devices, we have cluttered the system and made it a massive piece of software. No! Because of the structure of the system itself and the provision of the hooks that I referred to earlier, a particular user for the CBI system can select the device necessary for his own application and hook into the main CBI system through a small interface, which is provided at the request of the user.

Internal Interfaces

In considering the internal interfaces necessary for the total concept of the CBI system, we used as a guiding principle "adaptability to a specific user's instructional installation". Because of the wide variety of computer hardware throughout the academic and instrustrial training installations, we felt that a CBI system that could be built with the instructional capabilities and controls necessary for the total approach to computer-based instruction should also address itself to implementation on a wide variety of computer hardware. Therefore, the design of the system was constructed in order that the software package itself could be implemented on either large scale computer systems or on mini computer systems, regardless of computer manufacturer. It also should be able to support a number of kinds of terminals, and it should not be dependent on specific computer languages.

The I/O devices supported in the initial system development fall into three general categories, but the system need not be viewed as limited to these categories. At present, the types of terminals supported are: TTY compatible terminals; IBM 2741 compatible terminals; and IBM 3270 compatible terminals. These categories represent both hardcopy and CRT terminals produced by a number of different manufacturers. Again, to adapt to other types of terminals requires only small modifications to the appropriate internal interface structure.

The capability to support IBM 3270 type terminals allows for special screen formatting and cursor control. This special screen formatting will make it possible to place actual forms on the screen and let students fill in the appropriate sections. It also allows for a student's response to be logically placed in the answer to a question. For example, for the display

$$2 \times \_\_ = 8$$

a 4 will be entered by the student in the blank, making his total answer read

$$2 \times \underline{4} = 8$$

on the screen before him.

INTERFACE CAPABILITY

Two types of interfaces need to be considered in designing the total CBI system: (1) the interfaces necessary for the use of a variety of types of training aids in the instructional process; and (2) those interfaces necessary to use the CBI system on a variety of types of hardware. The interfaces for training aids are called "external" interfaces; those dealing with teleprocessing systems or operating and/or computing systems are called "internal" interfaces. Let's address the external interfaces first.

External Interfaces

In designing the CBI system, we looked at the instructional processes in a number of applications of industrial and academic training. Based on the results of this study, we determined that the training aids can be classified into those that we refer to as audio/visual aids and those that we refer to as graphic aids. I should clarify this because the classifications may seem to be overlapping.

Graphic aids, as we define them, are devices with features for drawing lines or for utilizing line displays. These devices may be either external to or incorporated into the CBI system hardware.

The use of graphics within the CBI system is possible depending on the terminal hardware connected to the system. Therefore, software hooks have been provided in the system to support the graphic function if the application requires that function and if the hardware supports it. For example, if an installation requires animated graphic capability and has the device available to support the function, then it is a matter of providing a small amount of interface to accommodate the function. This same theory has been used in the design of hooks for all of the audio/visual and graphic interface capabilities.

In addition to the use of graphics, we have provided the ability to hook into or to interface with graphic displays on various types of plotters. Again, the use of this function depends on the system application.

The system software is designed to utilize external audio/visual devices such as 35 mm slide projectors, audio records, video tape recorders, 35 mm audio/sync units, and phonetically generated voice response units -- almost any type of training aid currently available on the market. The important aspect about the use of these aids is that the control of the device may be programmed into the lesson by the instructor.

SYSTEM SUMMARY

I have partially described to you a very intricate and comprehensive, yet simple-to-use, computer-based instructional system. What will be the impact of this type of CBI system in the education and training field? To have any impact, several factors and considerations, other than functional design, had to be addressed.

First, costs were considered. Published statistics show that traditionally administered instruction costs range generally from $6 to $20 per student hour. For a CBI system to be cost effective, it must be functional at a competitive cost. Our tests have indicated that, based on experience of instructional developers, we can achieve a cost ratio competitive with traditional instruction.

Second, because of the wide range of capabilities, intricate data storage, and adaptability to a variety of hardware, care had to be taken not to impact installation core requirements or response time. These problems we have been able to prevent. The system is designed to operate within the boundaries of 16K plus OS and TP requirements, using 100K bytes of disk space. The actual requirements will depend on the extensiveness of use by the installation. Response time will fall in the average range of 0 - 2 seconds 90% of the time.

Finally, we must address acceptability by the user. And, frankly, the basic underlying current for success of any CBI system will be the development of good instruction by course authors. The author must play the key role in the success of the system. So the development of the CBI system must address acceptability especially by the instructor.

What motivates an instructor to write good CBI material? Three elements are to be considered: (1) comprehensive training and guidance in planning and constructing CBI material; (2) instructor control over materials and techniques to be used in specific training tasks; and (3) ease of system use. As a major part of our development effort, we have addressed and are accomplishing these goals.

So ladies and gentlemen, we at Boeing Computer Services present to you our concept of a total computer-based instruction system.

ANSWER ANALYSIS (AAABB)

CHARACTER ANALYSIS (AAABB1) - Deciding the answer status based on investigation of each character within the student response.

1. EXACT ORDERED - Each character in the response must be an exact duplicate of the anticipated response and must be given in the same order as in the anticipated response.

2. EXACT UNORDERED - Each character in the response must be an exact duplicate of the anticipated response but not necessarily give in the same order as in the anticipated response.

3. SCANNED ORDERED - Each answer is scanned character by character and comments/branches are allowed based on where the actual responses differ from the anticipated response. The characters are correct only if in the exact order as given in the anticipated response.

4. SCANNED UNORDERED - The same as #3 except the characters are correct is given regardless of which order in this technique, the breakdown occurs related to the omission of a specific character, rather than where it occurs.

5. TABBED EXACT - The same as #1 except selected characters may be omitted (tabbed over) from the check. A special character must be defined as a tab indicator.

6. TABBED SCANNED - The same as #5 except that comments/branches may vary depending on where the breakdown occurs.

7. EXACT ORDERED SUBSET - Same as #1 except the author may specify that the response if correct if a specified number of characters match.

8. EXACT UNORDERED SUBSET - Same as #2 except that the author may specify the response is correct if a specified number of characters match.

9. BNF NOTATION - Multiple characters may be considered to be correct for the same position within an anticipated response.

KEYWORD NOTATION (AAABB2) - Deciding the answer status based on an investigation of each word within the student's response. A word is defined as a series of characters delimited by blanks.

1. EXACT ORDERED - Same as #1 (AAABB1) except applied to keyword rather than character.

2. EXACT UNORDERED - Same as #2 (AAABB1) except applied to keyword rather than character.

3. SCANNED ORDERED - Same as #3 (AAABB1) except applied to keyword rather than character.

4. SCANNED UNORDERED - Same as #4 (AAABB1) except applied to keyword rather than character.

5. EXACT ORDERED SUBSET - Same as #7 (AAABB1) except applied to keyword rather than character.

6. EXACT UNORDERED SUBSET - Same as #8 (AAABB1) except applied to keywords rather than characters

7. BNF NOTATION - Same as #9 (AAABB1) except applied to keywords rather than characters.

8. PHONETIC SPELLING - The acceptance of phonetic equivalents in words. May be used with other appropriate analysis techniques.

Appendix 2

STANDARD SYSTEM REPORTS

Report 1: STUDENT LESSON SUMMARY

Method of Reporting: This report is given interactively to the student at the end of each lesson. The report is generated automatically.

Contents: The number of performances (questions) attempted by the individual student in that lesson, the number of performances judged correct, the number of performances judged incorrect, and the student's rating (grade/percent) on that lesson.

Report 2: STUDENT COURSE SUMMARY

Method of Reporting: Report is given interactively to individual student upon request by that student. The report can be requested at lesson end, or upon sign-in.

Contents: The courses completed by the requesting student, when completed, and the rating received by the student. In addition, any uncompleted courses scheduled for the requestor should be listed.

Report 3: STUDENT LESSON PATH

Method of Reporting: Called by instructional management function interactively.

Contents: For selected lesson(s), the path for a specified student through the instructional maze for each lesson -- identified by block ID's.

Report 4: STUDENT LESSON GRADE

Method of Reporting: Called by I M function interactively.

Contents: By student(s) by lesson(s)/course(s), the rating (grade/percentage) for the student.

Report 5: STUDENT COURSE PROGRESS

Method of Reporting: Called by I M function interactively.

Contents: This is an expansion of Report 2. In addition to Report 2 contents, it includes a breakdown of the data by lesson for one or more students as called.

Report 6: LESSON SUMMARY

Method of Reporting: Called by I M function interactively or batch. Call mode indicates output mode.

Contents: All activity for a specified lesson, including number of students, distribution of ratings, item response data.

Report 7: LESSON ITEM ANSWER

Method of Reporting: Called I M function in selective mode.

Contents: Item response data per each lesson item, including analysis mode, number of correct responses per item, number of incorrect responses, and number of unanticipated responses.

Report 8: LESSON GRADES (COLLECTIVELY)

Method of Reporting: Called by I M function by interactively/ batch mode.

Contents: Rating distribution by lesson.

Report 9: COURSE SUMMARY (BY LESSON)

Method of Reporting: Called by I M function in batch/interactive mode. Output is in selective mode.

Contents: Aggregated summary of Report 6, specified by course.

Report 10: NUMBER OF SYSTEM USERS

Method of Reporting: Called by I M function in batch/ interactive mode. Output is batch/interactive.

Contents: Number of system users classified by type of user, and by course/lesson breakdown (if applicable).

Report 11: SYSTEM COSTS

Method of Reporting: Called by I M function in batch modes; output is batch.

Contents: To be specified by individual user's system.

Report 12: ASSIGNMENT CODES

Method of Reporting: Called and output in batch mode from I M function.

Contents: Listing of code definitions and code assignments by type of user.